Gauss Jacobi:

A=input('Enter thr co-efficient matrix: ')

B=input('Enter Source Vector B: ')

P=input('Enter initial gauss vecotor: ')

n=input('Enter number of iterations: ')

N=length(B)

X=zeros(N,1)

for j=1:n

for i=1:N

X(i)=(B(i)/A(i,i))-(A(i,[1:i-1,i+1:N])\*P([1:i-1,i+1:N]))/A(i,i);

end

fprintf('Iteration no. %d\n,j',j)

X

if abs(X-P)<e

break ;

end

P=X ;

end

Gauss Seidel

A=input('Enter thr co-efficient matrix: ') ;

B=input('Enter Source Vector B: ') ;

P=input('Enter initial gauss vecotor: ') ;

n=input('Enter number of iterations: ') ;

e=input('Enter your tolerance: ') ;

N=length(B) ;

X=zeros(N,1) ;

Y=zeros(N,1) ;

for j=1:n

for i=1:N

X(i)=(B(i)/A(i,i))-(A(i,[1:i-1,i+1:N])\*P([1:i-1,i+1:N]))/A(i,i);

end

X

if(abs(Y-X)<e)

break;

end

Y=X ;

P=X ;

end

Trapezoidal:

f=input('Enter the equation: ');

x0=input('Enter the lower limit: ')

xn=input('Enter the number of intervals: ')

h=(xn-x0)/n ;

sum=f(x0)+f(xn) ;

for(i=1:n-1)

sum=sum+2\*f(x0+i\*h) ;

end

I=(h\*sum)/2 ;

Cramer’s rule

D = input('Enter your coefficient matrix: ') ;

b = input('Enter source vector: ') ;

N = length(b) ;

X=zeros(N,1) ;

d=det(A);

Aold=A;

if d~=0

for i=1:N

A(:,i)=b;

X(i)=det(A)/d;

A=Aold;

end

disp("Solution using cramer's rule is ")

X

else

disp('Cramer rule not applicable')

end